Team Developer 5.1 SP5 Release Notes.

Unicode Support

With the advent of Unicode support in TD5.1, many user applications using Sal APIs that deal with strings will need to be reviewed. For instance, TD automatically converts strings to Unicode when selecting data from database columns or reading strings from a file. Thus, complications may arise if the application code has dependencies on string sizes.

Example:

```
Function GetDataLength
Description: Get the data length in bytes without end separator
Returns
Number:
Parameters
String: sPar
Actions
Return SalGetBufferLength(sPar) - 1 ! Worked with older versions (i.e. before 5.1)
...
```

```
Actions
Set sInput = 'CAR'
Set nLen1 = SalGetBufferLength(sInput) ! Length is 8
Call SalStrToMultiByte(sInput,sOutput,ENC_UTF8)
Set nLen2 = GetDataLength(sOutput) ! Length is 3
If SalFileOpen(hFile, 'newfile.txt', OF_Create | OF_Write | OF_Binary)
Call SalFileWrite(hFile,sOutput,nLen2)
Call SalFileClose(hFile)
If SalFileOpen(hFile, 'newfile.txt', OF_Read | OF_Binary)
Call SalFileRead( hFile, sOutput, 100 )
Set nLen3 = GetDataLength(sOutput) ! length is 4 because sAnsi has two trailing 0x00
bytes
Call SalFileClose(hFile)
```

The issue here is the fact that we only want the string "CAR" written and read from a file.

The above code does the write correctly but the read incorrectly.

In order to make the above code work, the function GetDataLength needs to know if the string passed in is a unicode or an ascii string.

If it is a ascii string, we subtract 1. If it is a unicode string, we subtract 2.